ECE247 Project Report of EEG Classification

Rong Jin
SID:505523609
rongjin98@g.ucla.edu

Zhujun Fang
SID: 004736308
floraqa@g.ucla.edu

Abstract

This project aims to optimize the classification of the electroencephalography (EEG) data[2] set provided by the Brain-Computer Interaction(BCI) Competition[1]. In particular, the performance of various architecture including CNN-only, RNN-only, and the stacked CNN and RNN architecture were compared. We found that with data preprocessing, the 4-block CNN structure achieved the best performance with 73.2% accuracy across all subjects on the test data, and the RNN only structure outperforms others on single-subject tests data.

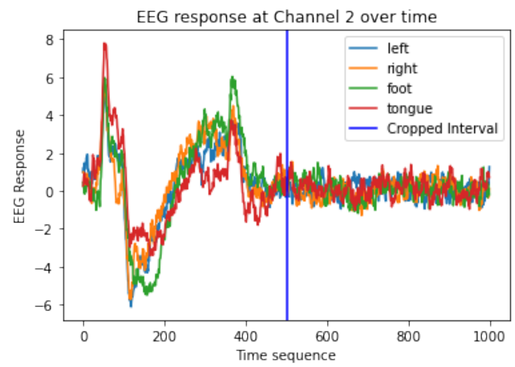


Figure 1. Data Visualization of the EEG Dataset

1. Introduction

In this project, the performance of three types of neural networks–CNN, RNN, and stacked CRNN were analyzed over both single-subject and all-subject dataset. To achieve the optimized performance, the effect of hyper-parameters, data-processing was further examined for all the networks.

1.1. Dataset

The EEG data was taken from nine individuals, with each measurement containing 1000 samples of EEG signals from 22 different electrode channels. Each set of data was classified into 4 different motor imagery tasks, namely the imagination of movement of the left hand, right hand, feet, and tongue. This makes the EEG dataset a 3D tensor with the shape of $[N \times C \times H]$, where N, C, W represents the number of data set, number of channels (22), and length of data set (1000) respectively.

1.2. Data Inspection & Pre-processing

To get a deeper inspection of the data, the time series of one particular channel (channel 2 in this work) of EEG signal for each imagery task were visualized as below. As shown in the figure.1, the first half section of signals has sharp curves, including both global maximum and minimum points which can be essential recognizing features, while the later half section of all signals flattens out and becomes less distinguishable. To improve the model performance, the EEG data was pre-processed based on figure.2, inspired by Tonmoy[3]. The raw data was firstly trimmed in

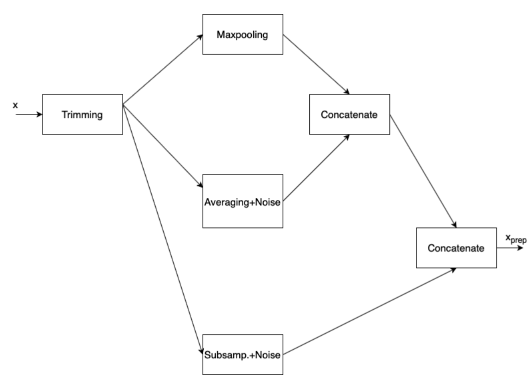


Figure 2. Data Preparation of the EEG Dataset

half, reducing the height to $H/2$. Then, Maxpool and averaging with white noise regulation was performed in parallel on trimmed data, that reduced height into $H/4$. Meanwhile, the data underwent a sub-sampling process which also reduced height by half but doubled the number of data. In the end, we concatenated the resulting data for augmentation and reshaped it into a 4D tensor, $[4N \times C \times \frac{H}{4} \times 1]$ as our input.

1.3. Model Overview

Convolutional Neural Network (CNN) The reshaped EEG data can be treated as 4N sets of images with height of $H/4$, width of 1 and channels of 22, which made the time series of EEG data available for CNN to analyze. The detailed structure of CNN was shown in the appendix 4.2.

Recurrent Neural Network (RNN) Since the electroencephalography (EEG) data is collected overtimes, RNN becomes an ideal model for this multi-classification problem intuitively. For the RNN architecture, we implemented **LSTM** and **GRU** respectively for performance comparison. Both LSTM and GRU contained 200 hidden dimensions and were bidirectional.

Hybrid Neural Network (CRNN) This network stacked the previous architecture of CNN and RNN together. CNN is advantageous at feature detection, extraction [4], and RNN enables the model to interpret time sequences. This hybrid neural network might give a better classification performance over the EEG data. CNN with LSTM and CNN with GRU were both examined in this project.

Therefore, five models were examined in total which were CNN, LSTM, GRU, CNN&LSTM, and CNN&GRU. For all the models above, the output features were connected into a dense neural network, which its structure can be found in appendix4.2. All the connected networks shared the same dropout probability. The cross-entropy loss was used for loss evaluation, and Adam with a learning rate 3×10^{-4} was used as the optimizer. The learning rate was automatically decayed once the average validation loss reaches a plateau. The size of the mini-batch was set as 64.

2. Experiment & Results

The results of model performance are splited into all-subject case and single-subject case separately.

2.1. All Subjects Performance

For all-subject cases, several naive trials were tested to check the general performance of each model on all-subject data. It was found that all five models suffered from over-fitting. In addition, CNN and CRNN models have significantly better performance than RNN-only models. Thus, hyper-parameters optimization was only performed on CNN and CRNN models.

1.Hyper-parameters The effect of dropout probability and the number of epochs was firstly examined on the CNN-only model. As shown in figure3, the dropout probability of 0.5 generally has better performance over other dropout probability, while a plummet of performance occurs when dropout = 0.8. In this trial, the best performance occurs at dropout = 0.5 and number of epochs = 70 with a test accuracy of 74.7%. Yet, it turned out this setting displayed a great fluctuation in performance, which makes the accuracy unable to be reproduced. As a result, dropout = 0.5 and epoch = 100 were chosen as the preferable setting, which gave an average test accuracy of around 72% to 73%.

From the previous results, dropout = 0.5 was picked for CRNN models by default. The effect of the number of stacks and the number of epochs was examined for CRNN models. From figure.4 and figure.5, the highest test accuracy for both CNN+GRU and CNN+LSTM occurs at 4 stacked layers of RNN and epochs fewer than 100 iterations. The final preferable settings of hyper-parameters on all five models are shown in the table.1

2.Train & Test Results The training and validation loss and accuracy with CNN and CRNN models are shown in figure.6 and figure.7. It can be noticed that all the models suffered from a severe training over-fit even with data augmentation. In fact, training data over-fitted even faster for the LSTM only and GRU only models, which over-fitted within 10 epochs. The test accuracy of 5 models with optimized hyper-parameters discussed previously is shown in the table.2.

3.Evaluation of Data Preprocess To evaluate the effect of data preprocessing on the model performance, another EEG dataset was collected with only data cropping. The performance of each model on both preprocessed and the cropped-only dataset is shown in the table.3. Each model was using the preferable hyper-parameters shown in the table.1.

2.2. Single Subject Performance

For the single-subject case, since the number of the dataset was reduced significantly, models that require many parameters such as CNN and CRNN will not be ideal since they inevitably over-fit the data. From the empirical results, the CNN model could only achieve an average of 45% test accuracy with preprocessed data, while two CRNN models could only achieve an average of 30% test accuracy. Therefore, only LSTM and GRU models are used for performance evaluation.

1.Test Accuracy of each subject The test accuracy of each subject was shown in the table4. Both LSTM and GRU models got the highest test accuracy on subject 9. Each subject data was preprocessed.

2.Evaluation of Data Preprocess Similar to the evaluation experiment for all-subject performance, another EEG dataset was collected with only data cropping. The performance of each model on both preprocessed and the cropped-only dataset is shown in the figure.8 and figure.9

3. Discussion

3.1. Evaluation of Data Processing

From the results, the data preprocessing improves the test performance of all models significantly for the all-subject dataset. Although the data preprocessing still improves the performance for single-subject cases in general, there do exist some subjects which data cropping suits better as shown in figure.8. It makes sense since max-pooling, averaging, and subsampling will inevitably cause the loss of information, which may break the continuity of the time serial EEG signals. For all-subject cases, the continuity of time sequences might be less significant since it is distorted by shuffling around different subjects, while the improvement from data augmentation by preprocessing weighs out the loss of information. However, when it comes to the single-subject case, the continuity of the signals might play a major effect in classification.

In addition, as the dataset was augmented by concatenating of the results of pooling and subsampling, the augmented data has a high correlation between each other which leads to another drawback of data preprocessing, that it made the model over-fit the validation data easily. In fact, from figure7, the validation accuracy is always higher than the corresponding training accuracy for all five models. Such results make the validation dataset no more an effective indicator of over-fitting, which makes the appropriate number of epochs hard to be determined.

3.2. Performance of Single-Subject Case

From table.4, the model performance varies significantly on different single-subjects. One possible explanation is that different subjects might have different time-span of active EEG signals. According to figure.1, the raw data was trimmed by half for both data preprocessing and cropping to discard the inactive EEG signals. However, for different subjects, one's active span EEG signals might not be exactly half of the time sequences. From the result data, we could surmise that 50% cropping suit the 9th subject, but might not be suitable for other subjects such as the first and the sixth subjects. Therefore, for future improvement on the single-subject dataset, the EEG data of each subject needs to be also visualized for more appropriate cropped data.

3.3. CRNN Performance

In the initial envision, the CRNN architecture should achieve the best classification performance among all the other structures for the all-subject dataset. However, from table.2, the additional RNN layers only undermined the performance of CNN architecture. One possible reason is that, during the forward propagation of CNN layers, the time length of the EEG data is reduced by a factor of $(\text{stride})^4$. For example, in this project, the stride of Maxpooling equals 3 and the time length of input data equals 250 due to data preprocessing. As a result, the output tensor only has a time length of 3 as the input of RNN layers, which conducts no useful time information but only worsens off the performance due to over-fitting. For future improvement, the architecture of CNN might need to be re-designed so that more timely information can be delivered to RNN layers.

4. Appendix

4.1. Plots

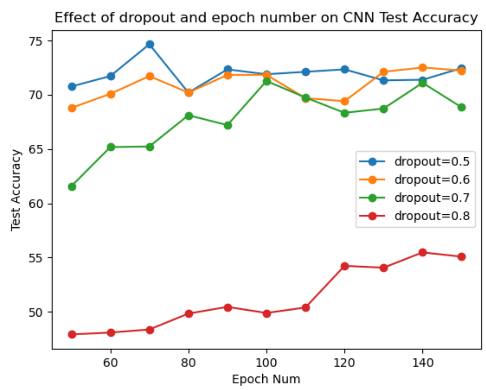


Figure 3. Effect of dropout and Epoch number on CNN Performance

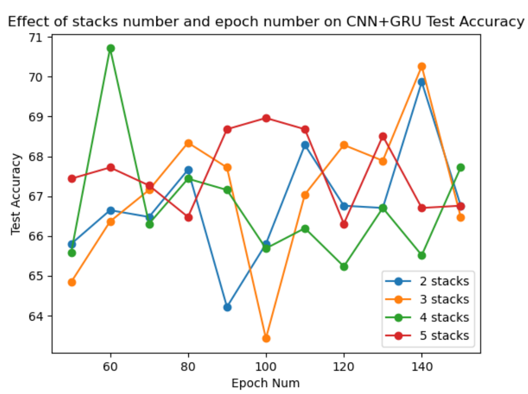


Figure 4. Effect of Stack number and Epoch number on CNN+GRU Performance

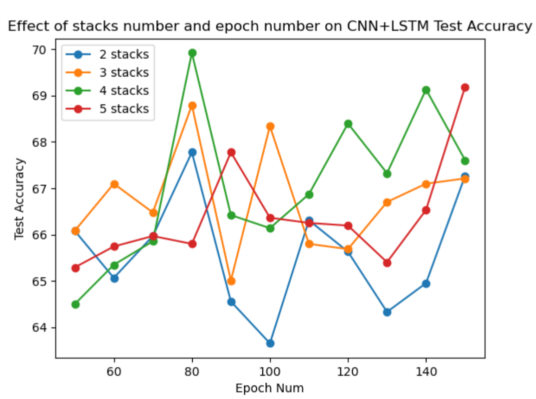


Figure 5. Effect of Stack number and Epoch number on CNN+LSTM Performance

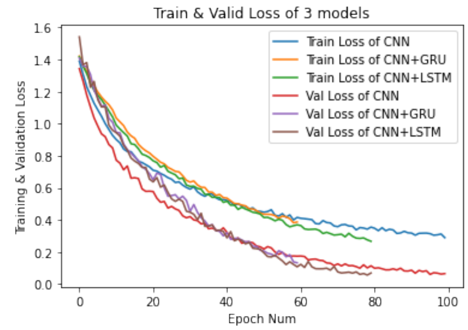


Figure 6. Training & Validation Loss of CNN CRNN models

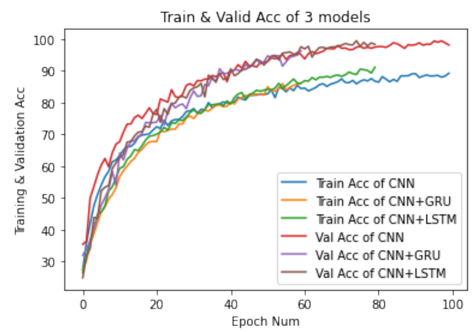


Figure 7. Training & Validation Acc of CNN CRNN models

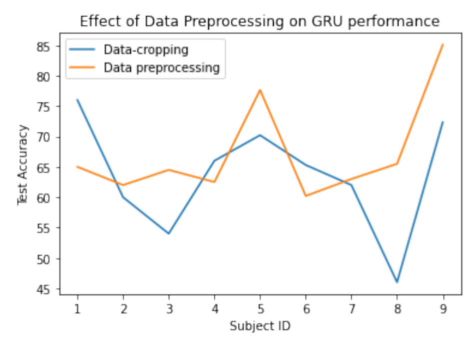


Figure 8. Effect of data preprocessing on gru

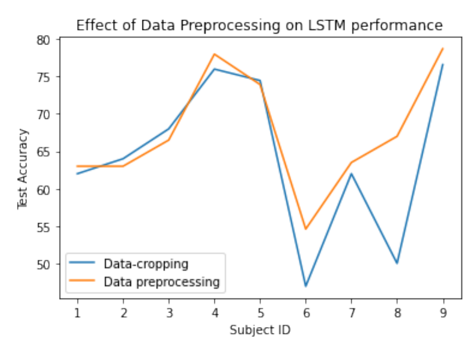
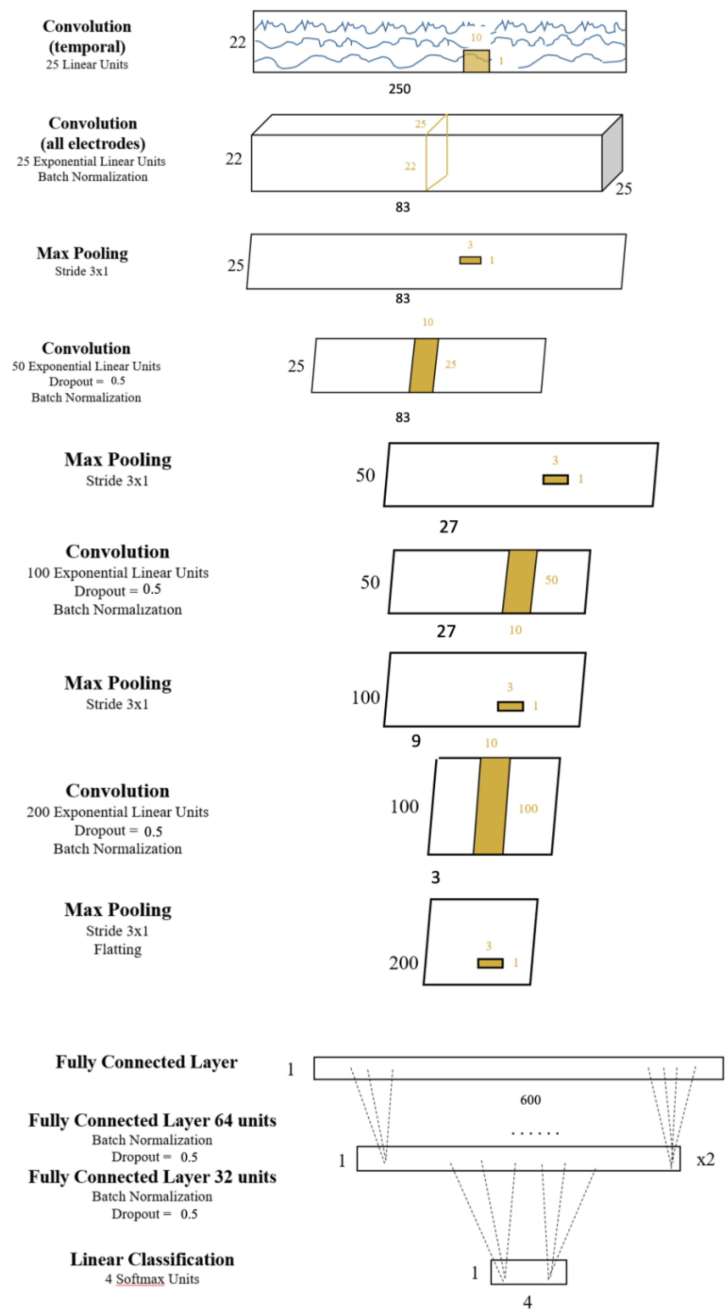


Figure 9. Effect of data preprocessing on lstm

4.2. Tables

[?]

Appendix Two: CNN Architecture



Model	Epoch	Stacked Layers	Dropout
CNN	100	-	0.5
LSTM	10	4	0.5
GRU	10	4	0.5
CNN+LSTM	80	4	0.5
CNN+GRU	60	4	0.5

Table 1. The choice of hyper-parameters

Model	Test Accuracy
CNN	**73.2**
LSTM	64.3
GRU	65.5
CNN+LSTM	68.1
CNN+GRU	68.2

Table 2. The optimized testing accuracy of 5 models

Model	Data Preprocessed Acc %	Data Crop Acc%
CNN	73.2	42.4
LSTM	64.3	57.3
GRU	65.5	56.7
CNN+LSTM	68.1	30.2
CNN+GRU	68.2	56.2

Table 3. Evaluation of effect of data preprocess on model performance

Subject ID	LSTM %	GRU %
1	63.0	65.0
2	63.0	62.0
3	66.5	64.5
4	78.0	62.5
5	73.9	77.7
6	54.6	60.2
7	63.5	63.0
8	67.0	65.5
9	**78.7**	**85.1**

Table 4. Test Accuracy of all single-subjects

References

[1] BCI Competition IV.

[2] T. B. B. group. motor imagery, uncued classifier application. https://www.bbci.de/competition/iv/desc_1.html, note = Accessed: 2022-3-10.

[3] T. Monsoor. Eeg data processing. https://bruinlearn.ucla.edu/courses/110319/files/8401084?module_item_id=4794338, note = Accessed: 2022-3-10.

[4] Z. Zhou, Q. M. J. Wu, S. Wan, W. Sun, and S. Xingming. Integrating sift and cnn feature matching for partial-duplicate image detection, 2020.